# Data Analysis and Solution seeking

# for finding the most popular courses

**Abstract**

In the sphere of web application developing and data mining, Python programming language has played a remarkable role so far. Because of many stunning features like regular expression, object-oriented, one line coding experience, cleaner syntax and easy to read or pick up, programmers are more likely to work with Python than Java, PHP and many other languages.

In this project we are looking forward to do some of the data mining work and accomplish some analytical researches jobs by writing python scripts, for the purpose of helping students who might want to find out the most popular courses on “www.myeducationpath.com”

**Target URI: http://myeducationpath.com/courses/**

Data element we have got from this website:

- Name of the course

- Category

- Description

- Provider

- Free or not

- Number of votes

- Similar courses

- Related books

- Related iTunes resources

**Research Questions:**

What are these popular courses from each category and what makes them more “popular” then other course?

Definition of “popular”:

Before we started to find out why they are popular, we will face a question: how to tell which course is popular. To answer this question, we consider those courses with high vote numbers and high vote values as popular courses. (Or maybe we can take into calculation of number of students as well)

**Methodology:**

For the convenience and efficiency of each and every query, as well as for the deduction of data redundancy, we decide to use Database to store all data sets we have previously collected. So for all sub tasks, we were using Mongodb Database to store all data set. Here we listed before the result charter all MongoDB scripts(written in JavaScript) that retrieves the data we need.

**Sub-tasks:**

**1. We analyze all those courses with high votes values, try to find the correlation between the specific category and the course’s popularity.**

*Following script will pool out all data combination of Category name, its average vote value, number of all courses within it, and average number of voters.*

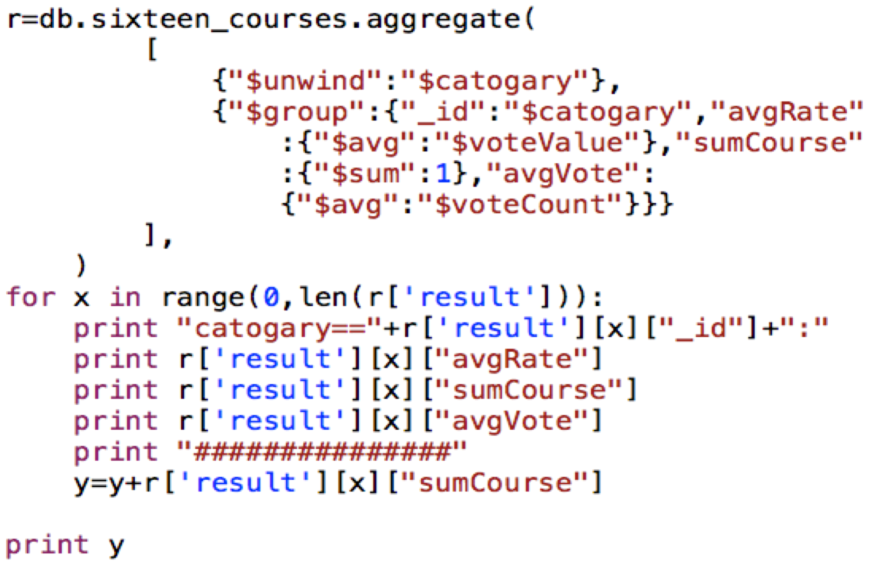


Diagram 1-1

From Diagram 1-1, we can see that courses from different categories have various average vote values. For example courses in categories like Computer Science and English Literature may have more popularity as well as the vote values. Meanwhile, some unpopular categories like General Interdisciplinary Studies and Health and welfare give rise to the unpopularity of courses in them. However, we cannot simply make the assumption only based on average vote values. In order to find out more, we drew another diagram based on the data we collected in Diagram 1-2.

Diagram 1-2

Here is the second diagram to show the correlation between categories and popularity from courses in them. Instead of showing the average vote values for each category, we summed up the total vote value for courses in each individual category (total vote value= average vote value \* number of votes). As we can see, Computer Science is still the leader in our list, but Health and Welfare becomes relatively unpopular after our calculation. The result from this step seems to be more convictive.

**2. We analyze all those courses with high votes values, try to find the correlation between their provider and the course’s popularity.**

*Following script will pool out all data combination of provider’s name, its average vote value, number of all courses within it, and average number of voters.*

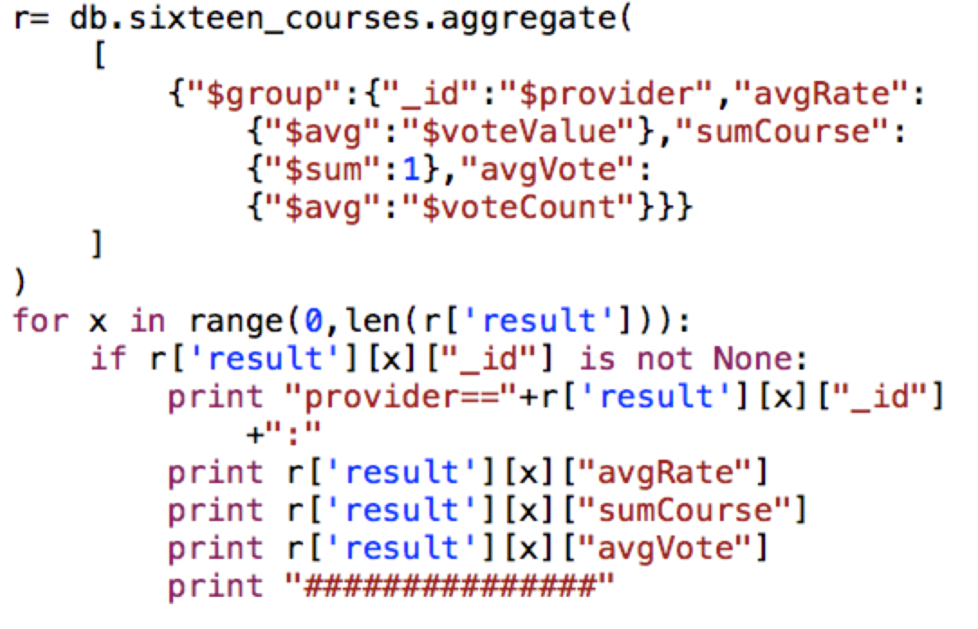


Diagram 2-1

From Diagram 2-1, we noticed that in all 400 high rating courses, over 90% of them were provided by two largest provider: Coursera and edX. So we want to know if courses by these two providers are more popular than others. To answer this question we come up with an idea of drawing another pie chart(Diagram 2-2). By checking the portions of vote number for each provider, the answer to this question, became clear.

Diagram 2-2

However, Diagram 2-2 shows that the number of voters for Coursera and edX appears to be less than Udacity and Udemy. Thus, although Coursera and edX provide more courses than other two providers, we still cannot consider them as more popular. In order to find out whether popular provider makes its courses popular, we shall compare the average vote value for these providers.

Diagram 2-3

After checking the average vote value for each provider, Coursera and edX are less than other two providers. This result implies that popular provider does not necessarily make their courses popular.

**3. We analyze the average vote value and number of votes of each category from 400 courses, from 800 courses and from 1600 courses to see if there was a trend or preference among different categories.**

*For this part, MongoDB queries code stays the same way as it was shown in first tasks. We did the same of this query for three times to find out if there were some fluctuations on average vote values from different scale of data.*

Diagram 3-1

We can see that some minor fluctuations truly existed when the scale of data increased. And the average vote values were getting closer as data scales up, which makes average vote value from 800(medium scale) and 1600(large scale) much more similar than that from 400(small scale). However, even if there did exist some fluctuation, some popular categories stayed the same, such as Computer Science, Mathematics, English literature, Social Science and Business.

**4.The correlation between vote value and number of similar courses. Also we are going to find out the relationship between vote value and number of related books:**

In order to find out if the number of similar courses has something to do with the vote value, we collected the data to draw the diagram below.

*Here is the MongoDB Script for this query:*

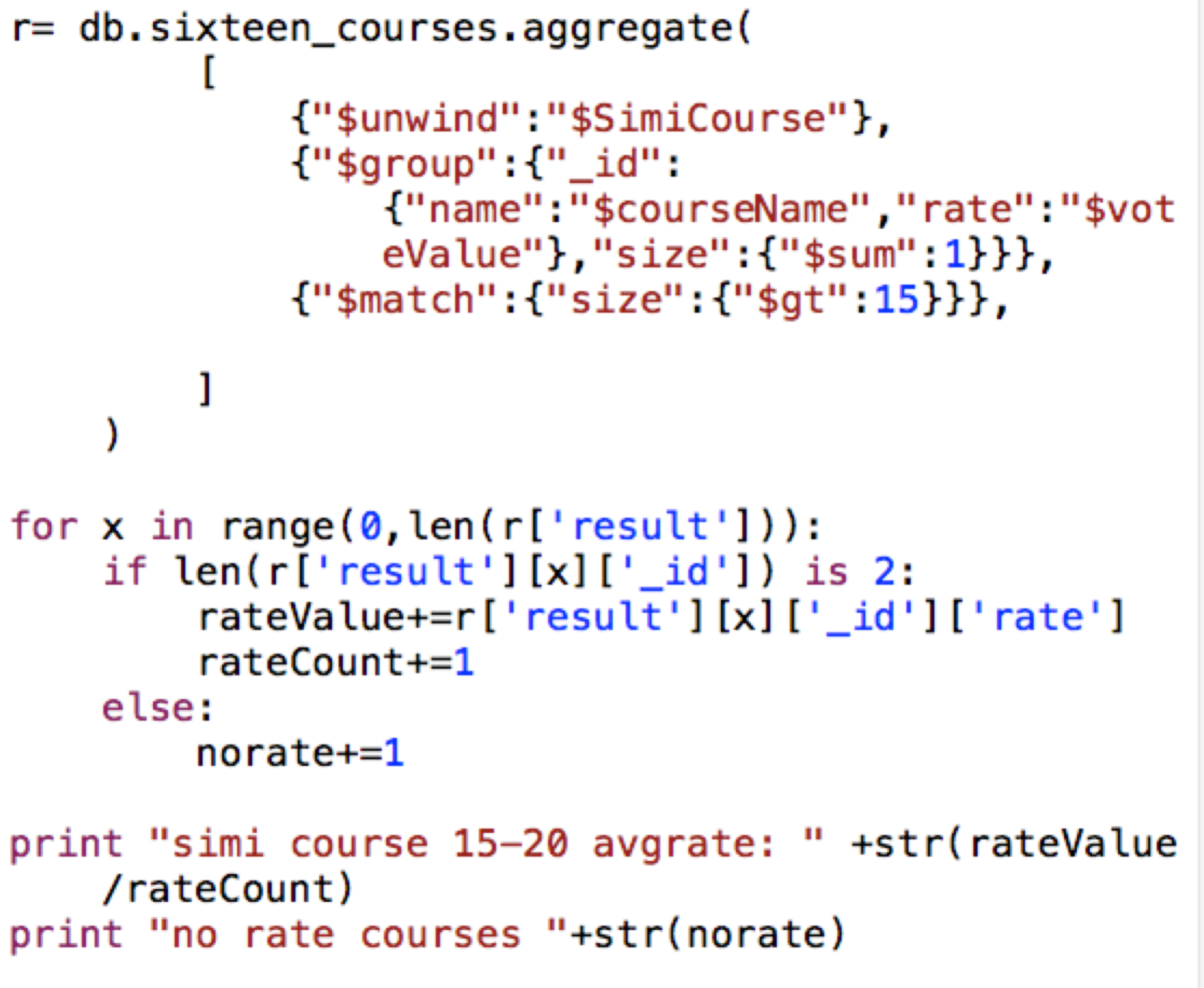


Diagram 4-1

In Diagram 4-1, average vote values of courses with more than 5 similar courses increase with the number of their related books. However, courses with less than 6 related books have higher average vote value than those who have 6 to 15 courses. So we cannot simply assume that average vote value depends on the number of related books.

Also we wanted to figure out the potential relationship between vote values and number of related books for each course. Here is how we collected data combining number of related books and courses average vote value from MongoDB:



Diagram 4-2

The Charter above(Diagram 4-2) shows that there was no certain prediction whether number of related books determined the average vote values.

So, next we decide to check if the popularity of a course has something to do with the existence of its related books. Here we drew a diagram of average vote value for courses with related books and those without any recommended book.

*Here is the MongoDB script for this query:*

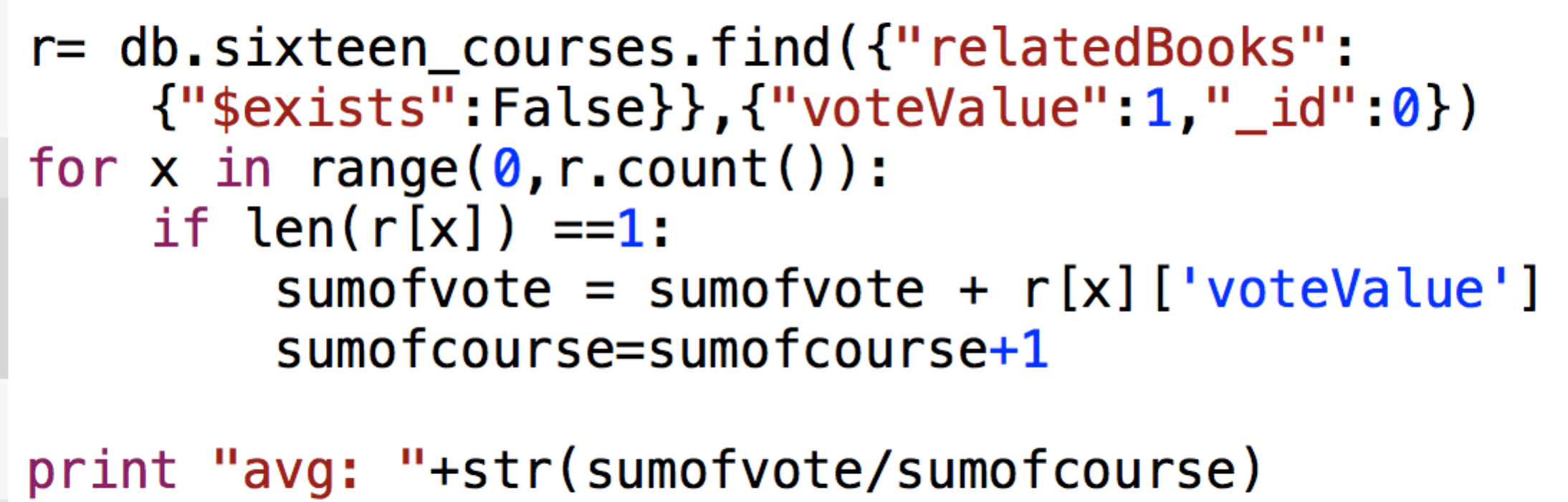


Diagram 4-3

After scrutinizing the data we have, we found courses with related books tend to be more popular than those without recommended books.

**Challenges:**

1. About the evaluation of descriptive text, since each individual course may have description from various perspectives, we found it hard to judge so large amount of text information by telling which piece of text means more popular than others.

2. This website has archived more than fourteen thousand courses. Even if we tried to download all possible information from all courses on this website, the downloading program will cost great amount of time. We tried to download 3200 courses by looping through the whole site, and it turned out that it took 4 hours.

3. Some of the course pages content (html content) were not written in one particular criterion, which makes the regular expression in our python code hard to search for some of our data sets.

**Overview of Our Data**

Vote value of 1600 courses:

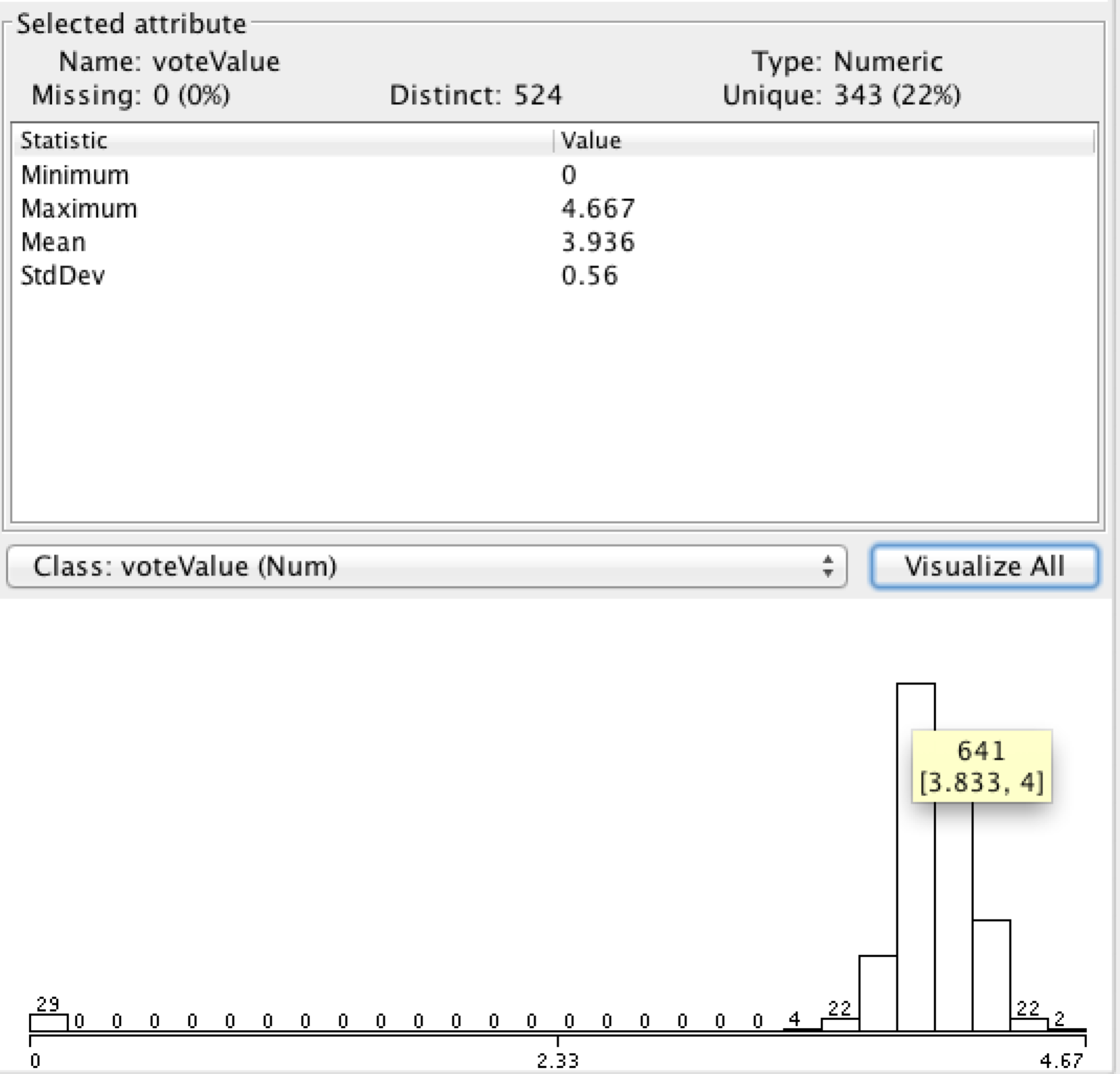
****

Diagram 5-1

As we can see from Diagram 5-1, majority of courses among this 1600 high rating courses have vote value around 3.8 to 4 point. And to be more specific, we list some statistical results below:

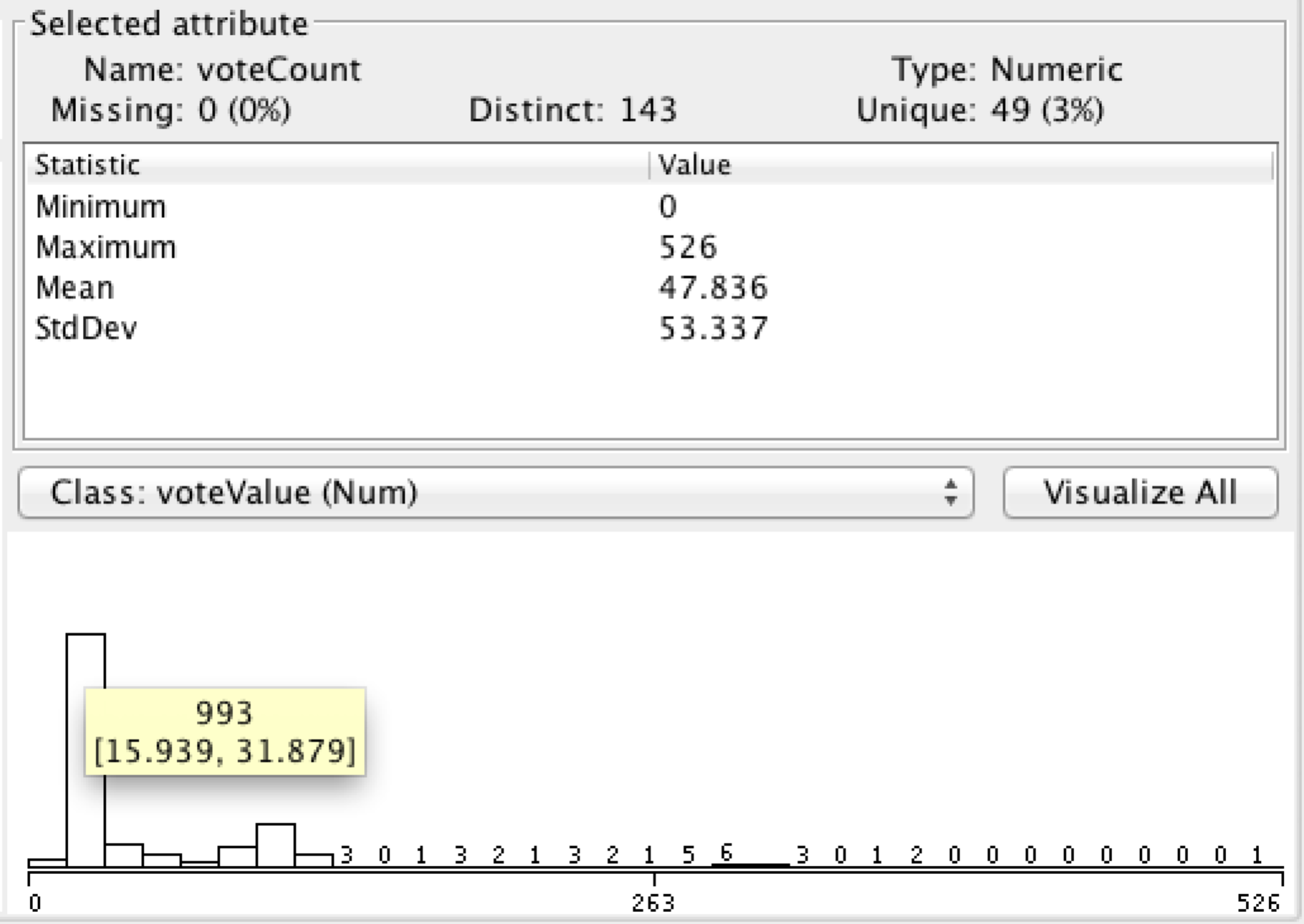
Minimum value: 0 Course vote intervals:

Maximum value: 4.667 [0,0.167]: 29 [3.5,3.667]: 22 [3.667,3.833]: 138 [3.833,4]: 641

Mean value: 3.936 [4,4.167]: 510 [4.167,4.333]: 206 [4.333,4.5]: 22 [4.5,4.677]: 2

Standard Deviation: 0.56

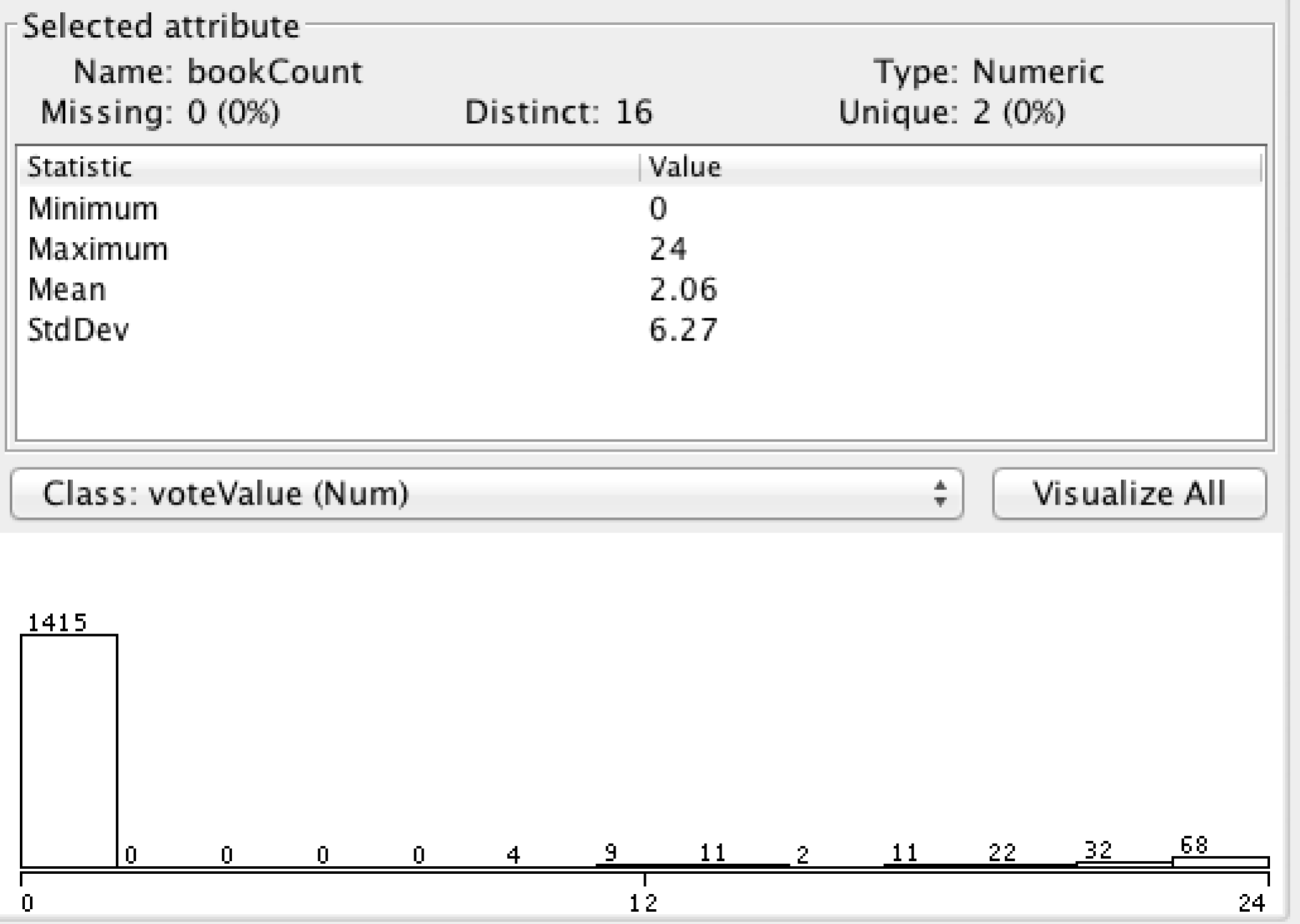
Course amount with different vote participants:



Number of Similar Courses:



Number of Related books:

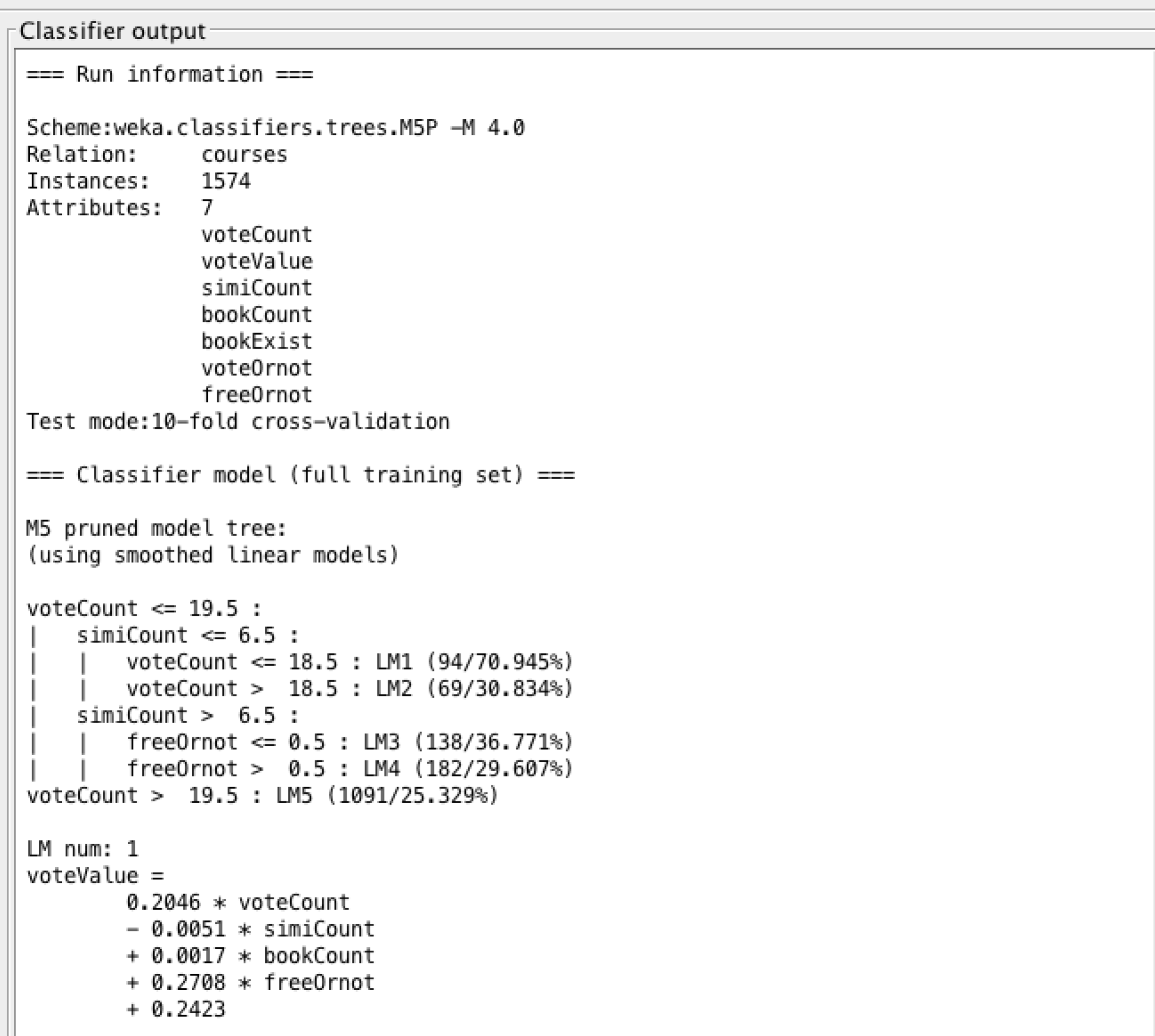


Tagclouds for Most Popular Courses:

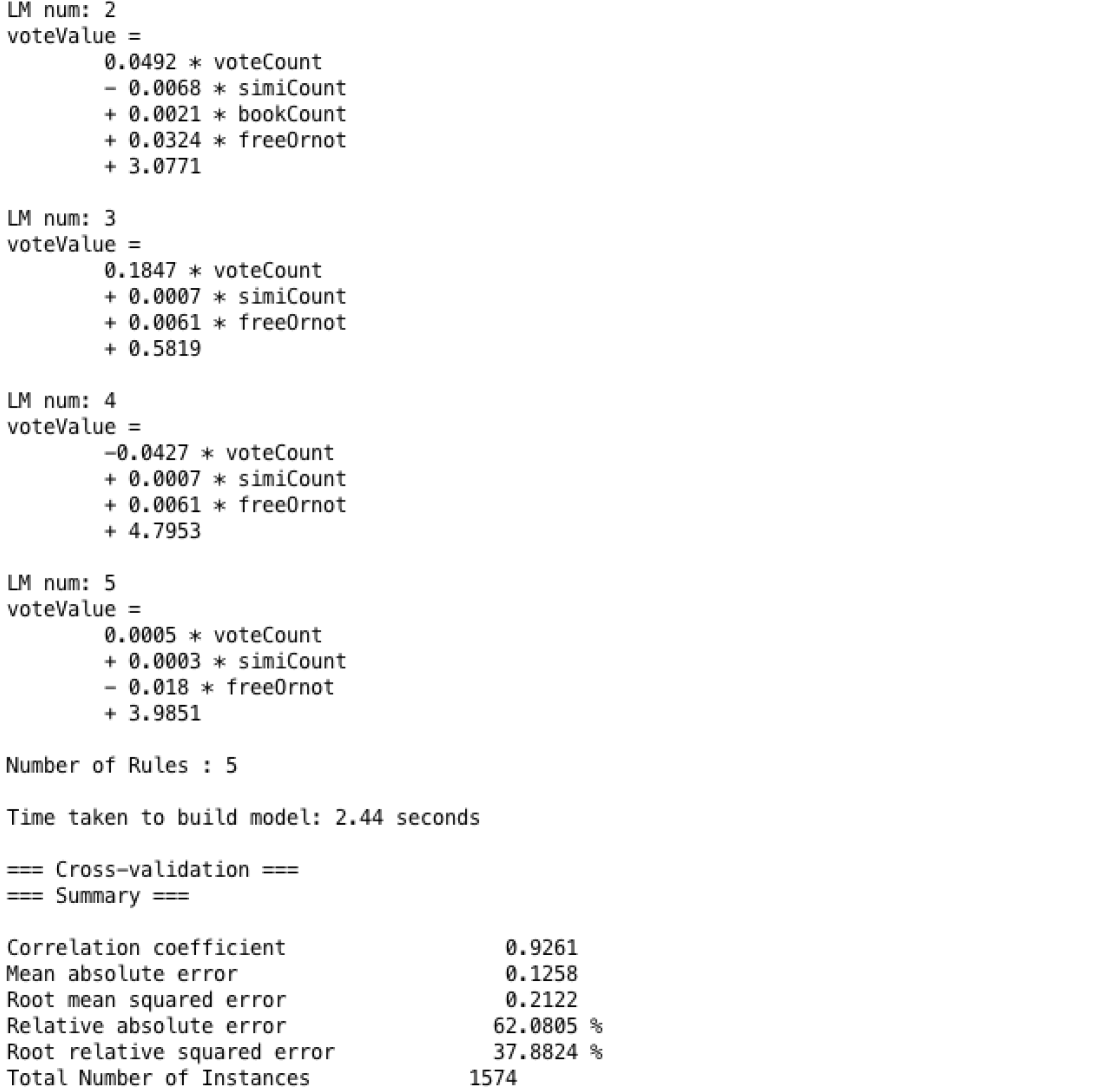


**Using WEKA to make a decision tree**

After reviewing data sets, we wrote “1600courses.arff” and load this file with data mining software WEKA to generate a Decision Tree.



According to the output, we got 3 attributes related to the vote value. They are number of votes, free or not (1 or 0), and number of similar courses. This decision tree can help decide the vote value of a certain course.



Here we can see the correlation coefficient is 0.9261, which means the this decision tree will make prediction well.

**Conclusions:**

1. Computer Science is overall the most popular category on our target URI.

2. Free courses got more reputation than those paid courses.

3. Courses with related books and resources is more attractive to students.

4. Rating value does not depend on how many similar courses that course has.